APL-Journal

A Programming Language

1/2005

APL-Germany e.V.

Jahrgang 24 Nr. 1 2005

ISSN - 1438-4531

RHOMBOS-VERLAG

In dieser Ausgabe:

Christiane Görges

Wer testet, ist feige, wir tun es dennoch

APL-Gegentestwelt zum produktiven C-System bei einer großen deutschen Lebensversicherung

Helmut Mumbauer

Anbindung von OpenGL an Dyalog-APL

Komplexität der grafischen Möglichkeiten

Michael Baas

"Schöne APL-Programme" – mal anders

Mit Dyalog Workspace als .EXE „kompilieren"

Im Blickpunkt

Tagungsbericht

Dyalog User Conference 2005 in Elsinore

www.apl-germany.de

INHALT

Seite 4

Dyalog User Conference 2005

Die diesjährige Dyalog Benutzer-Konferenz vom 19-21. Oktober 2005 fand statt in Elsinore, Dänemark (ca. 1:30h Zugfahrt von Kopenhagen), im Hotel "LO Skolen" (Labour Organisation School), einem von einer Gewerkschaftsstiftung betriebenem Hotel, in dem hauptsächlich Schulungen stattfinden, das aber auch Privatpersonen offen steht.

Seite 7

Wer testet, ist feige

APL-Gegentestwelt zum produktiven C-System bei einer großen deutschen Lebensversicherung

Seite 9

Anbindung von OpenGL

In diesem Beitrag wird gezeigt, mit welcher Technik OpenGL-Funktionen in Dyalog-APL zu integrieren sind. In Beispielen wird die Komplexität der grafischen Möglichkeiten dargelegt.

Seite 11

Versionsmanagement mit APL

Für ein erfolgreiches Arbeiten in Projektteams ist es erfoderlich das Standards definert werden. Diese Standards beziehen sich auf...

Seite 12

"Schöne APL-Programme"

Mit Dyalog.net bzw. Version 10 führte Dyalog die Möglichkeit ein, einen Workspace als .EXE zu „kompilieren" (bzw. eine .EXE zu erstellen, in die der Workspace eingebunden ist und mit Hilfe des Interpreters in einer .DLL ausgeführt wird.)

Nachrichten Seite 16

Der digitale Organismus

Karlsruhe. Computer sind vielseitig. Sie führen Verkaufsverhandlungen, organisieren Auktionen oder übernehmen die Steuerung von Maschinen.

Impressum 2

EDITORIAL 3

Dyalog User Conference 2005 in Elsinore 4

Wer testet, ist feige, wir tun es dennoch 7

Anbindung von OpenGL an Dyalog-APL 9

APL4CVS Versionsmanagement mit APL + CVS 11

"Schöne APL-Programme" – mal anders 12

Die kleine Maus hat keinen Schwanz 14

Nachrichten 16

Anzeige 20

Impressum

ISSN 1438-4531

Herausgeber: Dipl.-Math. Dieter Lattermann, Walldorf

Redaktion: Dipl.-Volksw. Martin Barghoorn (verantwortlich)
Technische Universität Berlin
Franklinstr. 28
D-10587 Berlin
Tel. (030) 314 24392
Fax (030) 314 25901
eMail: Martin@Barghoorn.de

Erscheinungsweise: halbjährlich

Erscheinungsort: Berlin

Verlag: RHOMBOS-VERLAG, Berlin
Kurfürstenstr. 17
D-10785 Berlin
Tel. (030) 261 9461
Fax (030) 261 6300
eMail: verlag@rhombos.de,
Internet: www.rhombos.de

Satz: RHOMBOS-VERLAG, Garip Sari

Druck: dbusiness.de GmbH, Berlin

Copyright: *APL Germany e.V.* (für alle Beiträge, die als Erstveröffentlichung erscheinen)

Fotos: M. Barghoorn (Titel, Seite 4), Michael Baars (S. 4 und 5)

Homepage des APL-Germany e.V.: http://www.apl-germany.de
Homepage des APL-Journals beim Verlag: http://www.rhombos.de/shop/a/show/article/?99

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung von Redaktion und Herausgeber nicht übernommen werden. Sämtliche Veröffentlichungen im APL-Journal erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Mit Namen gekennzeichnete Artikel geben nicht unbedingt die Meinung des Herausgebers oder der Redaktion wieder. Für unverlangte Einsendungen wird keine Haftung übernommen. Nachdruck ist nur mit Zustimmung des Herausgebers sowie mit Quellenangabe und Einsendung eines Beleges gestattet. Überarbeitungen eingesandter Manuskripte liegen im Ermessen der Redaktion.

Wie immer freuen wir uns auf Ihr Feedback zu diesem Heft. Schreiben Sie uns an barg@cs.tu-berlin.de, oder faxen Sie an: 030 - 314 25901.

Technische Hinweise für Autorinnen und Autoren

Manuskripte werden von der Redaktion entgegengenommen. Sie müssen im Besitz der Nutzungsrechte für die von Ihnen eingereichten Texte, Fotos und Zeichnungen sein. Die Zustimmung zum Abdruck wird vorausgesetzt.

Manuskripte und Beiträge aller Art werden in jeder Form entgegengenommen. Es werden jedoch Beiträge in gespeicherter Form bevorzugt. Bitte speichern Sie Ihren Text im Format Ihrer Textverarbeitung und zusätzlich als RTF-Datei (Rich Text Format). Sie brauchen sich nicht die Mühe machen, Ihren Text aufwendig zu formatieren. Auszeichnungen im Text sollten sich auf „Kursiv“ beschränken. Ihre Datei schicken Sie der Redaktion bitte auf einem Datenträger (Diskette oder CD-ROM). Alternativ hierzu können Sie der Redaktion Ihre Daten auch per Email (Martin@Barghoorn.de) schicken.

Wenn Sie Ihr Manuskript einreichen, sollten Sie zusätzlich eine Gliederung und Zusammenfassung Ihrer Arbeit vorlegen.

Wichtig: In der Zusammenfassung (abstract) sollte in wenigen Sätzen die wichtigste Aussage Ihres Textes wiedergegeben sein.

Grafiken

Bitte senden Sie uns einen sauberen Ausdruck der Graphiken auf rein-weißem Papier. Linien (keine Haarlinien!) und Schriften in den Graphiken sollten so groß gewählt sein, dass sie auch bei (stark) verkleinerter Wiedergabe im Heft noch zu erkennen sind; auf der Rückseite vermerken Sie bitte Ihren Namen und die Nummer der Abbildung; Zusätzlich wäre es schön, wenn Sie uns die mit einem gängigen Vektor-Grafikprogramm erstellten Zeichnungen und Grafiken auch als Datei liefern. Grafikdateien (Strichzeichnungen) sollten möglichst im Austauschformat EPS (skalierbar) mit 300 bis 600 dpi Auflösung in schwarz-weiß bzw. Graustufe abgespeichert werden. Nach Rücksprache können sie zusätzlich Ihre Grafiken in Ihrem Original-Dateiformat liefern oder in ihr Textdokument einbetten.

Digitale Fotos benötigen wir im TIF-Format mit einer maximalen Auflösung von 250 dpi. Nicht geeignet sind Abbildungen und Fotos, wenn Sie aus dem Internet stammen beziehungsweise im Original als Bildschirmdarstellung (Auflösung 75 - 150 dpi) vorliegen. Ebenfalls nicht geeignet sind Grafiken, die mit einem pixelorientierten Zeichenprogramm erstellt wurden.

Liebe APL Freundinnen und APL Freunde,

im Mai diesen Jahres habe ich von Dieter Lattermann das Amt des Vorsitzenden von APL Germany übernommen. Für das Vertrauen, welches mir von den Mitgliedern entgegengebracht wurde, möchte ich mich recht herzlich bedanken. Ich selbst habe APL im Jahre 1975 bei dem Wissenschaftlichen Zentrum der IBM in Heidelberg kennengelernt. Später habe ich APL im Bereich des Prototyping von Bankanwendungen benutzt. Große Erfahrungen mit Sharp APL habe ich als Systembetreuer für einige Bankanwendungen sammeln können

Nicht versäumen will ich, mich bei dem bisherigen, langjährigen Vorsitzenden für seine gute Arbeit und sein Engagement zu bedanken, mit dem es möglich gewesen ist, stets gute Referenten zu Vorträgen für die APL Germany Treffen und Beiträge für das APL Journal zu finden.

EDITORIAL

Damit dies so bleibt, sind alle Mitglieder und APL Interessierte aufgerufen, sich an APL Germany mit Beiträgen an den APL Tagungen und dem APL Journal zu beteiligen. Das Team im Vorstand tut sein bestes, die Aktivitäten von APL Germany zu organisieren und tatkräftig zu unterstützen. Am Leben gehalten wird APL Germany aber von den Beiträgen der Mitglieder. Nur gemeinsam ist es möglich, APL Germany weiterzuführen.

Reiner Nussbaum

Dyalog User Conference 2005 in Elsinore

Veranstaltungsort: Die diesjährige Dyalog Benutzer-Konferenz vom 19-21. Oktober 2005 fand statt in Elsinore, Dänemark (ca. 1:30h Zugfahrt von Kopenhagen), im Hotel "LO Skolen" (Labour Organisation School), einem von einer Gewerkschaftsstiftung betriebenem Hotel, in dem hauptsächlich Schulungen stattfinden, das aber auch Privatpersonen offen steht. Das Hotel liegt etwas abseits vom Ort in einem großen Parkgelände mit Golfplatz und reizvollen Wanderwegen. 2 bis 3-geschossig gebaut fügt sich der Hotel-Komplex gut in die Landschaft ein und war mit seiner interessanten und abwechslungsreichen Architektur auch als Schaustück dänischer Handwerkskunst konzipiert, was auch gelungen ist – der Chronist jedenfalls war begeistert von den immer neuen Blickwinkeln und es entstand eine Fülle nun zu sichtender und zu bewertender Fotos.

Durch die etwas andere Finanzierung des Hotels und der Gehälter kann beim Einkauf von Rohmaterial für die Küche sehr großzügig kalkuliert werden, und so ist das LO Skolen berühmt für die gute Küche –nach übereinstimmender Meinung der Teilnehmer auch zu Recht. (Es wurde gescherzt, die Teilnehmer würden nicht Kilobytes mit nach Hause nehmen, sondern Kilogramm.)

Organisation

Markenzeichen einer guten Veranstaltungsorganisation ist, dass man nichts davon bemerkt – und in diesem Sinne war die Organisation ausgezeichnet. Zu bemängeln wäre lediglich der Regen am Donnerstagnachmittag – dem Tag des „Wikinger-Tests", der im Freien statt fand – aber das kann nicht wirklich der Organisation angelastet werden.

Incl. Dyalog/Insight-Personal waren 99 Teilnehmer anwesend, neuer Rekord für Dyalog.

Als Mitbringsel gab es das obligatorische T-Shirt, Tasche und Tasse sowie einen Memory-Stick mit allen Vorträgen und einer weiteren alpha-Version (nach Naples) von V11.

Vorträge

Die Vorträge boten eine bunte Mischung aus technischen Inhalten unterschiedlicher Schwierigkeitsgrade über Demonstration von Anwendungen und Vorträgen zum Management von Software-Projekten bis hin zum Singen fröhlicher Lieder („APL Blossom Time"). Das Programm war weitgehend in einem Strang geplant, so dass man selten zwischen zwei interessanten Vorträgen wählen musste.

Mittwoch

Der Schwerpunkt des ersten Tages waren „Neuigkeiten". Es wurde nochmals auf die geänderten Eigentumsverhältnisse bei Dyalog Ltd. eingegangen und die zukünftige Planung sowohl in Bezug auf Eigentumsverhältnisse wie auch Entwicklung des Unternehmens skizziert und, so ganz nebenbei, ein neues Logo und die neue CI von Dyalog eingeführt. Zusammenfassend sieht es nach „neuem Schwung" bei Dyalog aus (obwohl Dyalog eigentlich nie den Eindruck erweckte, einen „Schwungmangel" zu haben) und es sind viele Initiativen geplant, um die Nutzung und Bekanntheit von APL auszuweiten. (U.a. gibt es nun für Hochschulen/Universitäten kostenlose Lizenzen.)

Es wurde angemerkt, dass für einen Programmierer die Lösung für ein Problem wohl immer auch Programmierung zu beinhalten scheint, während man nun erkannt hat, dass die weitere Verbreitung von APL nicht mehr von neuen Features abhängt, sondern eher Schulungsmaterial usw.

Natürlich wurden die Neuerungen von Version 11

vorgeführt, die nach Möglichkeit noch vor Dezember fertig gestellt werden soll. Eine Alpha-Version wurde den Teilnehmern bereits ausgehändigt. Neben Spracherweiterungen (Function-Power-Operator, Domäne von UND und ODER erweitert auf Integers zur Berechnung von kleinstem gemeinsamem Vielfachen bzw. größten gemeinsamem Teiler) standen vor allem die objektorientierten Erweiterungen im Vordergrund. Mit V11 wird es möglich, „Klassen“ zu erzeugen, die Methode und Eigenschaften haben, und über diese „Schnittstellen“ genutzt werden können, ohne dass man sich um Details der Implementierung (....¨¨kümmern müsste. Details, die nur kurz gestreift wurden, sind ein neuer „Trigger“-Mechanismus, der bei Änderungen von Variablen ausgelöst wird sowie die Möglichkeit, im internen Editor Ereignisse zu behandeln, wobei nicht klar wurde, ob ausser Speichern+Beenden noch weitere Ereignisse unterstützt werden. (In der nächsten Ausgabe des APL-Journals werden die Neuerungen in einem separaten Artikel ausführlicher vorgestellt.)

Weiterhin ist eine 64-bit Version in Entwicklung, deren Hauptmerkmal die Erweiterung des Adressraums auf 64bit und damit der Wegfall praxisrelevanter Grössenbeschränkungen für Workspaces (derzeit: ca. 1.7GB), aber auch für die Größe von Komponentendateien und Variablen ist. Die 64bit-Version wird unabhängig von der 32bit-Version vertrieben und es soll keine kostenlosen Runtime-Lizenzen geben. Geplante Verfügbarkeit: kurz nach V11, vermutlich Anfang 2006.

Am Abend wurde die Fernsehsendung „The Origins of APL“ (aus 1974) mit Ken Iverson, Adin Falkoff, Larry Breed und Roger Moore wiederaufgeführt. Frisuren und Kleidung der Akteure sorgten für Heiterkeit, vorausschauend und noch immer aktuell war aber die Antwort von Roger Moore auf

die Frage, wo die Schwerpunkte der künftigen Entwicklung liegen würden: „Performance, Interfacing with other systems“. (Dyalog erwägt, standardmäßig dieses Video bei künftigen Konferenzen vorzuführen, wenn die Rede auf neue Features kommt ...)

Donnerstag

Nach einem 1,5-stündigen Tutorial von Morten Kromberg zu „OO in APL“ wurden am Donnerstag im Wesentlichen Anwendungen diskutiert, etwa FlipDB von Paul Mansour (www.flipdb.com), eine vollständig in D implementierte Datenbank. Auch TakeCare, die bereits 2004 in Naples präsentierte Anwendung zur Verwaltung von Patientendaten schwedischer Krakenhäuser (mehr als 1Mio DCF-Files mit Patientendaten in einer C/S-Lösung mit über 18.000 Anwendern) stand erneut im Blickpunkt, als ein zwischenzeitlich für TakeCare entwickeltes Interface präsentiert wurde, welches TakeCare-Funktionalität und Daten über Webservices zur Verfügung stellt.

Am Nachmittag fand dann der „Wikinger-Test“ („Viking-Challenge“) statt, bei dem 4 Teams in 4 Disziplinen (Axt werfen, Messer werfen, Luftpistole und Pfeile blasen) gegeneinander antraten. Unter Leitung eines martialisch anmutenden „Survival-Teams“ fanden diese Prüfungen in den Tiefen der dänischen Wildnis statt. Nach anfänglich gutem Wetter stellte anschließender Regen die Wasserfestigkeit der Teilnehmer auf die Probe, doch schließlich überstanden alle Teilnehmer den Test lebendig und mit viel Spaß. Die Ergebnisse sollen demnächst veröffentlich werden und neben der offensichtlichen Prämierung der Gewinner wurde bereits angekündigt, dass es Zusatzpreise geben soll für die besten statistischen Auswertungen des Ergebnisses und die originellsten Berechnungsvorschläge für alternative Ergebnisse.

Weitere Themen der Vorträge am Donnerstag waren die statistische „Qualitätskontrolle“ an der University of Wales, der Einsatz statistischer Methoden beim Umgang mit großen Datenmengen und ein Suchen+Ersetzen-Werkzeug, welches mit regulären Ausdrücken arbeitet.

Das Bankett am Abend war ein erneuter Beweis der außerordentlichen Qualität der Küche.

Freitag

Um nach der kurzen Nacht den Kreislauf wieder in Schwung zu bringen, demonstrierte ich kurz mein Programm zum „Beamen“ von Liedtexten, welches gewöhnlich im Gottesdienst eingesetzt wird. Begleitet von Paul Mansour auf der Gitarre sangen die Teilnehmen dann das Lied „APL

Blossom Time“ mit (angesichts der frühen Uhrzeit) doch erstaunlicher Begeisterung. Danach führte Stefan Lanzavecchia in einem weiteren Kurzvortrag vor, wie man die Google Desktop-Suche benutzen kann, um APL-Code zu durchsuchen.

Schwerpunktthema war dann das Management von Software-Entwicklung. Die Vorträge von Klaus Løie-Andersen, der die Vorgehensweise von Simcorp (Über 100 Entwickler, eigene Abteilung „Produktionsplanung“) schilderte („Wasserfall“-Methode, Spezifikationen, detaillierte Vorgaben, Erfolgskontrolle, Metriken, viel Statistik und permanentes Feedback) und Stephen Taylor (Entwicklung „taktischer“ System bei größter Versicherung Englands in kleinen Teams in unmittelbarer Zusammenarbeit mit den Anwendern, ohne „klassische“ IT-Beteiligung, keine schriftlichen Unterlagen zur Projektbeschreibung („der Code ist die Spezifikation“)) deckten wohl die beiden Extrempunkte des Spektrums möglicher Vorgehensweisen ab. Doch beide sind auf ihre Art erfolgreich – stellt sich die Frage, wann man von der einen Vorgehensweise auf die andere umschwenken muss.

Am Nachmittag sorgte Brooke Allen mit seinem Vortrag „How to be lucky“ noch für die philosophische Komponente, die ein rundum gelungenes Meeting abschloss.

In der Abschlussrunde, die ich wegen des frühen Rückflugs leider versäumte, sollten die Schwerpunkte der weiteren Entwicklung diskutiert werden, dabei wurden von Dyalog in einem zuvor ausgeteilten Whitepaper Stichpunkte angeboten wie etwa Datentypen (Unicode), Rationale und komlexe Zahlen, Source-Code Management, Ausbildung und Dokumentation, Werkzeuge zum Datenzugriff und Skalierbarkeit. Erste Schritte in die gewünschte Richtung werden erst nach der Auslieferung von Version 11 gegangen und wer noch Wünsche/Beiträge/Anforderungen zur weiteren Entwicklung hat und die Konferenz versäumte, ist eingeladen, Morten Kromberg (mkrom@dyalog.com) zu kontaktieren.

Fazit

Neben Hotel, Landschaft und Vorträgen war es auch sehr interessant, viele alte Gesichter wiederzusehen, aber auch neue Gesichter und die dazugehörigen Personen kennenzulernen und so konnte ich interessante Kontakte knüpfen. Zusammenfassend empfand ich die Konferenz als rundum gelungen und im Vergleich zur Konferenz 2004 (in Naples, USA) wie auch zu den vorigen APL-Konferenzen als die beste und ich bin mit Sicherheit in 2006 wieder dabei.

Zitate

Auf der Tagung aufgeschnappt

Eine kleine Sammlung von Zitaten, frei übersetzt ins Deutsche:

- “Um Alzheimer zu vermeiden, soll man, wenn man älter wird, weiterhin Neues lernen. Daher werden wir auch weiterhin jedes Jahr ein neues APL-Primtiv einführen, um Alzheimer bei APL-Entwicklern zu vermeiden.“ (Morten Kromberg)
- „Es ist vermutlich nicht besonders beeindruckend, bis man darüber nachdenkt.“ (John Scholes)
  „Warum verschwendet dieser Referent wertvolle Trinkzeit?“ (John Scholes)
- „Diskutieren ist besser als Schnarchen.“ (John Scholes)
- „Eine meiner unzüchtigen Phantasien ist, dass eine junge hübsche Dame an der Bushaltestelle mich anspricht und fragt: ‚Entschuldigung, aber kennen Sie den Unterschied zwischen einem Baum und einer Liste?’“ (John Scholes)
- „Ich hasse Computer!“ (John Scholes)
- „Das ist niedlich, aber potentiell nutzlos.“ (John Daintree)
- „Der Entwickler, der das geschrieben hat, ist in Schwierigkeiten. Besonders in Ländern, wo es noch die Todesstrafe gibt.“ (Dan Baronet)
- „Ich möchte kein C++ schreiben. Ich hab’s in der Vergangenheit getan und es tut weh.“ (Stefano Lanzavecchia)
- „Die 2 grossen Lügen in der Software-Entwicklung: 1. Ich kann Ihnen sagen, was wir brauchen. 2. Ich kann Ihnen sagen, was es kostet.“ (Stephen Taylor)
- „Kommunikation ist unzuverlässig und teuer. Als Ingenieur sollte man die Verwendung unzuverlässiger und teurer Werkzeuge minimieren.“ (Stephen Taylor)
- „Man braucht keinen Raketenwissenschaftler, um eine Anwendung in APL zu entwickeln. Ein Nuklearforscher reicht auch.“ (Veli-Matti Jantunen)
- „Möchtest Du das Geschirr vor oder nach dem Dessert spülen?“ (Brooke Allens Grossmutter)
- „Man sagt, es gäbe zwei Arten von Rednern: diejenigen, die Lampenfieber haben, und solche, die lügen.“ (Brooke Allen)

Wer testet, ist feige, wir tun es dennoch

APL-Gegentestwelt zum produktiven C-System bei einer großen deutschen Lebensversicherung

MOTIVATION:

Kleine Fehler – Großer Verlust

- Lebensversicherung mit einer Leistung von EUR 100.000,-
- Rückkauf nach 25 von 30 Jahren Laufzeit
- Garantiert: Deckungsrückstellung
- Nicht garantiert: Überschüsse
- Einheitssumme: EUR 1.000,-

Annahme: Einheitswert fehlerhaft

- (100.000 x (EW+a)) / 1.000
- (100.000 x EW) / 1.000 + (100.000 x a) / 1.000
- a=1 EUR ergibt eine Differenz von 100 EUR
- Einheitswerte liegen jeglichen Berechnungen zugrunde!

HISTORIE:

HOST-System

- PL/1-Bestandssystem auf dem HOST
- APL-Abgleiche auf dem HOST

Außendienstsoftware

- Außendienstsoftware auf dem PC in C++
- APL-Abgleiche auf dem PC

► Produktivsysteme auf dem HOST und PC ohne Zusammenhang. APL-Systeme mit einem gemeinsamen Rechenkern.

AUSGANGSSITUATION:

Situation im Frühjahr 2003

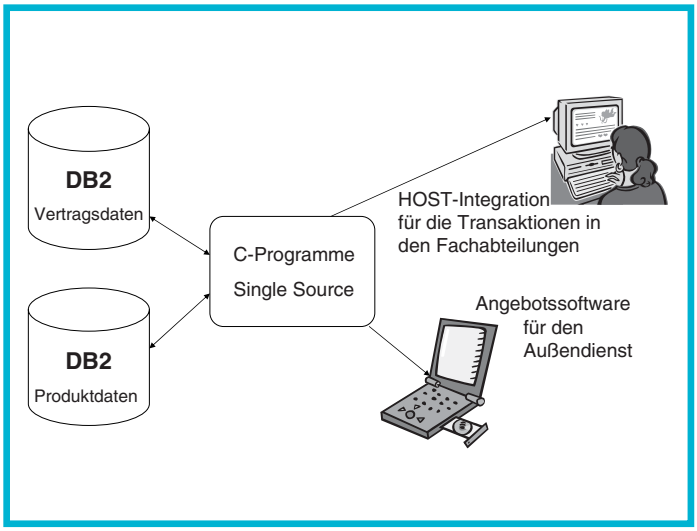
- Projekt zur Entwicklung des neuen Produktivsystems in der zweiten Hälfte – geplante Fertigstellung seinerzeit 1.7.2004
- Abgleichsystem noch offen
- Gesetzliche Änderung des Garantiezins bereits zum 1.1.2004
- Neue Produkte im neuen System bereits zum 1.1.2004 zu implementieren

Testsystementwicklung in APL

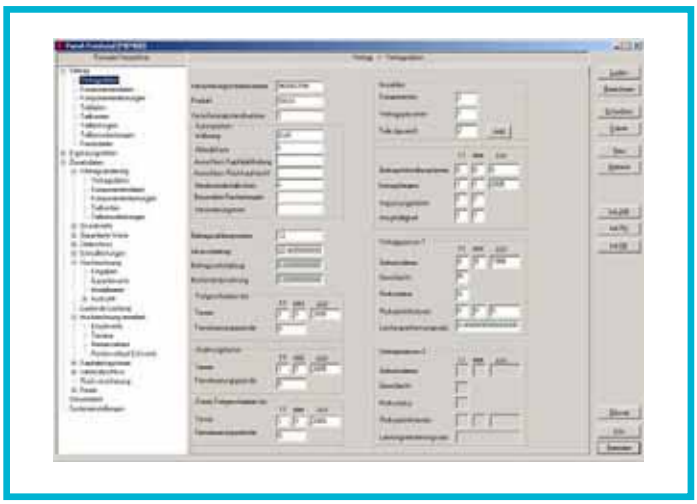
- Ansatz: Bestehende Programme des damals geltenden Tarifwerks als Basis
- Ohne endgültige Definition der Schnittstellen wurde mit der Umsetzung des Tarifwerks begonnen
- APL erwies sich als sehr effizienter Weg
- APL Know How innerhalb der Abteilung wurde genutzt und soll stetig erweitert werden

SYSTEMKOMPONENTEN UND VORGEHENSWEISE:

Produktivsystem (PS)



Frontend zur Testunterstützung und Vertragsansicht



Vorgehensweise bei der Implementierung neuer Produkte

Produktivsystem:

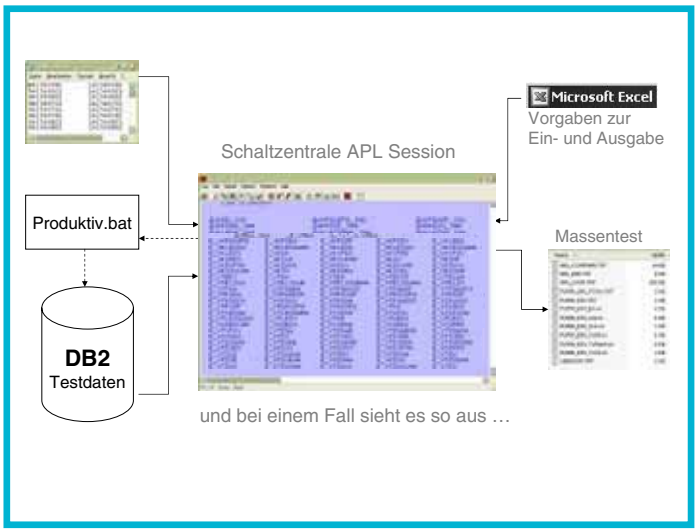
- Geschäftsplan
- Fachkonzepte
- Umsetzung in C

Testsystem:

- Geschäftsplan
- Direkte Umsetzung in APL

TESTPROZESS UND -KOMPONENTEN

Wie wird getestet?

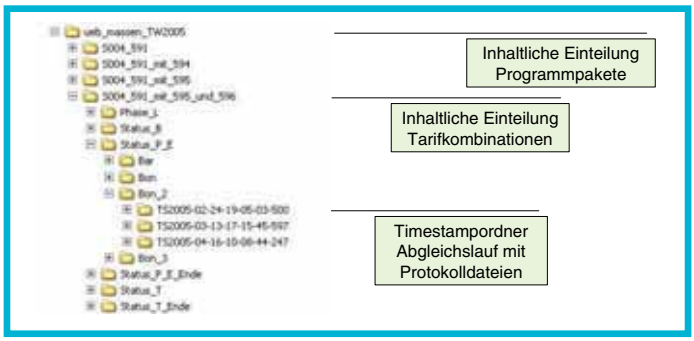


Pluspunkte im Testprozess

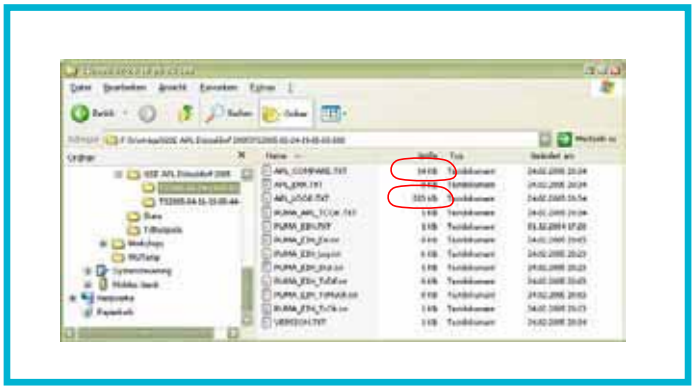
- Übersichtliche Darstellung und Ordnung des Testfall bestandes
- Statistische Aufbereitung der auftretenden Fehler
- Ansicht und gleichzeitige Dokumentation sämtlicher Fehler auf einen Blick
- Effizienz bei der Fehleranalyse

APL -TESTSYSTEM IN AKTION

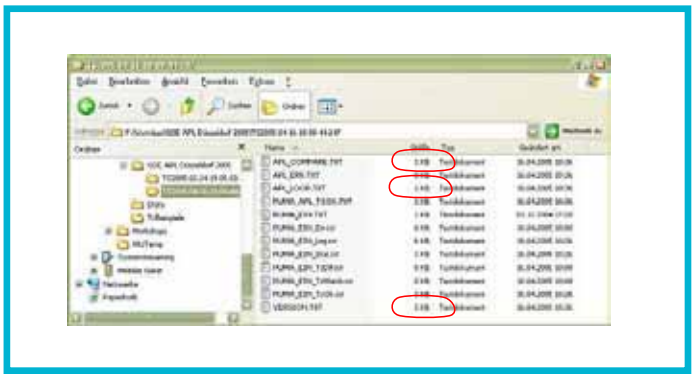
Verzeichnisstruktur im Netz



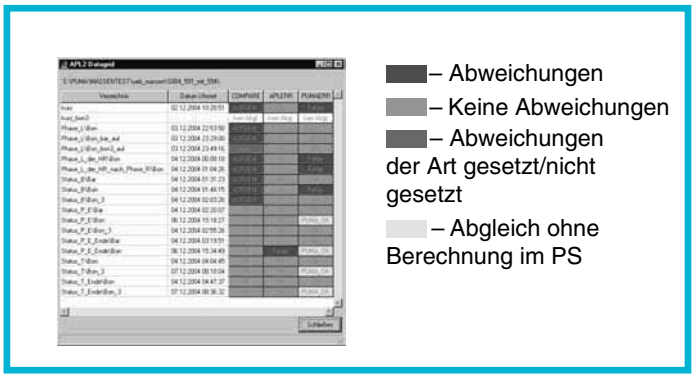
TS-Ordner mit Abweichungen



TS-Ordner ohne Abweichungen



Im Massentest sehr hilfreich …



WEITERE FEATURES

Was sonst noch möglich ist

- Automatische Übernahme von Datenmodelländerungen
- Kopieren von Testfällen innerhalb der DB2-Datenbank
- Auswertung von nächtlichen Läufen, die das Produktiv system intern durchführt
- neue Ideen werden regelmäßig umgesetzt – die Produkte zum Herbst 2005 und Januar 2006 werden integriert

Kontakt:
Christiane Görges
DPC GmbH
23. Mai
GSE Düsseldorf

Anbindung von OpenGL an Dyalog-APL

In diesem Beitrag wird gezeigt, mit welcher Technik OpenGL-Funktionen in Dyalog-APL zu integrieren sind. In Beispielen wird die Komplexität der grafischen Möglichkeiten dargelegt. Insbesondere wird auf die Performance bei Interaktionen (Event-Handling) eingegangen. Ein Highlight der Applikation ist die Identifikation der OpenGL-Graphik.

Was bietet OpenGL:

- 3D-Ausgabe
  - schattierte Punkte, Linien, Polygone, Flächen
  - Stacks für Attribute und Matrizen
  - Ausblenden verdeckter Objekte
  - Arbeit mit Display-Listen
- Materialeigenschaften, Beleuchtung, Texturmapping
- Verschiedene Buffer für Spezialeffekte
- Identifikation von Objekten

OpenGL arbeitet unabhängig vom Betriebssystem und vom verwendeten Window-System. Daraus folgt, dass diese Library nur für die eigentliche Darstellung, also das Rendering einer Szene ausgelegt ist und daher auch nicht über Interaktionsmechanismen verfügt. Hier setzt Dyalog APL auf. Mit Hilfe des GUI (Graphical-User-Interface), dem dort enthaltenen Event-Handling wird aus dem reinen Display-Mechanismus ein interaktives System.

Technik der Anbindung einer OpenGL-Funktion

In Dyalog-APL werden OpenGL-Funktionen mit Hilfe der APL-Systemoperation: []NA (Name Association) angebunden. Hierbei wird der Zugriff auf kompilierte Funktionen einer Dynamic Link Library(DLL) ermöglicht.

Im nachfolgenden Beispiel soll gezeigt werden, wie OpenGL-Funktionen in Dyalog-APL angebunden werden:

```
'glVertex3f_DLL' []NA 'OpenGL32.DLL.C32|glVertex3f F4 F4 F4'
```

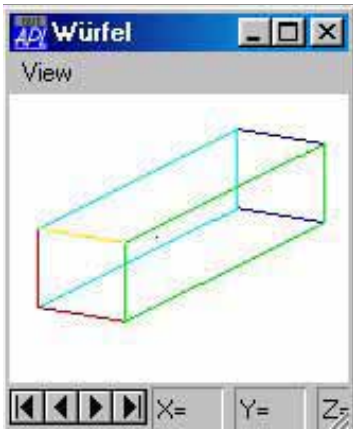
Voraussetzung ist, dass die Opengl32.dll-Library im Systemverzeichnis zur Verfügung steht. Die Syntax der []NA-Anweisung sieht als linkes Argument einen APL-Funktionsnamen vor, das rechte Argument ist wie folgt aufgebaut: [result] library|function [arg1] [arg2] ...

Der Zusatz: .C32 sagt aus, dass es sich um eine 32-bit Library handelt und der Aufruf nach C-Konventionen abläuft. Die Argumente F4 beschreiben den Datentyp (F für float) und die Datengröße (4-byte). Nach der Assoziierung steht die Funktion: glVertex3f_DLL zur Verfügung. Als Parameter können die Koordinaten von Punkten übergeben werden.

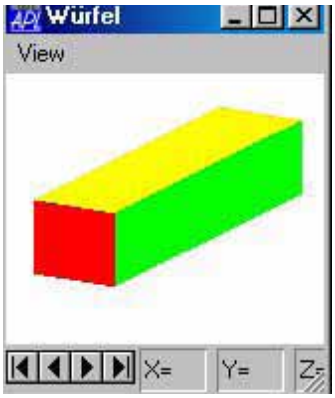
Beispiele für erzeugte Grafik in 3D

Grafische Ausgaben als Drahtmodell:

Dargestellt ist ein Drahtmodell, die Eckpunkte sind als Vertices definiert. Wird vor Definition der Vertices der Parameter GL_QUAD_STRIP gesetzt, so bilden die Vertices einen Streifen aus aneinandergrenzenden Vierecken, wobei ein neues Viereck aus den alten und zwei neuen Punkten generiert wird.



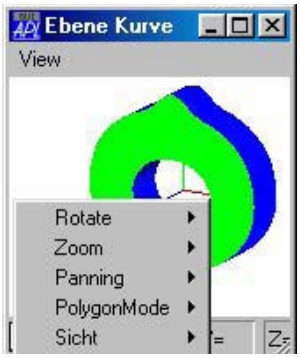
Grafische Ausgaben als Flächenmodell



Dieser dargestellte Würfel ist aus dem Drahtmodell abgeleitet. Durch Aufruf der Funktion glPolygonMode und dem Parameter GL_FILL entsteht dieses Flächenmodell.

Grafische Interaktion:

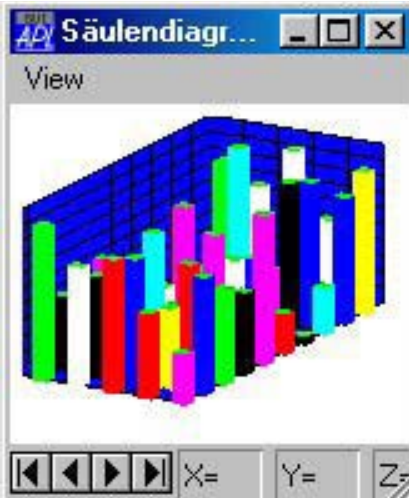
Rotating, Zooming und Panning



Zur Unterstützung von Visualisierungen stehen Rotate-, Zooming und Panning zur Verfügung. Wichtig ist bei 3D-Betrachtungen die Möglichkeit, einen Körper von allen Seiten zeigen zu können. Der aktuelle Modus wird in der Statusleiste angezeigt.

Als weitere Hilfsmittel sind vordefinierte Ansichten vorhanden. Es kann zwischen den Ansichten: xy xz yz und der xyz-Darstellung gewechselt werden.

Grafische Identifikation



Dieses Beispiel zeigt ein Säulendiagramm. Die Besonderheit hier ist, dass jede Säule gezielt identifiziert werden kann. Damit ist es möglich, komplexe Datenbestände grafisch darzustellen und Fragen an das Datenmodell zu stellen. Der Betrachter ist somit nicht beschränkt auf die Ablesegenauigkeit an der Achsenbeschriftung, sondern kann genaue Informationen über jede Säule erfragen.

Performance der Grafik

Die Performance der Grafik muss differenziert betrachtet werden:

a) Performance bei der Erzeugung der grafischen Objekte

OpenGL ist vom Datenmodell betrachtet punkte-orientiert. Dies bedeutet, dass die Geometrien in Form von Polygonen beschrieben werden. Wird dies in der klassischen Art und Weise „Punkt für Punkt“ angegangen, so ist die grafische Erzeugung des OpenGL-Datenmodells sehr langsam. Zur Verbesserung dieses Übergabeproblems stehen im Anwendungsworkspace: OpenGL2APL.ws (E-Mail: Helmut.Mumbauer@t-online.de) Funktionen zur Verfügung, die die Performance bis zu einem Faktor 10 verbessern. Im Detail geschieht dies durch Matrizen, die von APL-Funktionen in den Hauptspeicher geschrieben werden und durch OpenGL-Funktionen übernommen werden.

>getAddress - Zuordnen von Speicherplatz

>writeAddress - Beschreiben von Speicher

>glDrawElements - OpenGL-Funktion zur Übernahme von Matrizen

>freeAddress - Freigeben von Speicherplatz

b) Performance bei Interaktion

Bewertungskriterien für die Performance der Interaktion sind: Bei der Betrachtung des Modells durch Rotation, sollte der dargestellte Körper der Cursorbewegung unmittelbar folgen. Ferner sollte die Bewegung ruckfrei und flackerfrei sein.

Zur Arbeitsweise der Interaktions-Funktionen am Beispiel Rotation: Während der Programmausführung führt jede Verschiebung der Maus zu einem Event in APL. Dadurch wird die aktuelle Transformationsmatrix neu berechnet, an OpenGL übergeben und es entsteht eine neue Darstellung.

Fazit: Das Zusammenwirken von Dyalog-APL und OpenGL erfüllt die genannten Kriterien und ist für den praktischen Einsatz sehr gut geeignet.

Anbindung von OpenGL an Dyalog-APL

Ziel von OpenGL ist es, ein Software-Interface zur Hardware zu sein. Zweck ist die Darstellung von zwei- und dreidimensionalen Objekten mittels eines Bildspeichers. Diese Objekte sind entweder Images, die aus Pixeln zusammengesetzt sind, oder geometrische Objekte, die durch Vertices (Raumpunkte) beschrieben sind.

Die Ursprünge von OpenGL gehen auf die Graphic Library IRIS GL (IRIS war der Name einer Workstation, GL ist die Abkürzung für Graphics Library) zurück. Maßgeblichen Anteil an der Entwicklung hatte Silicon Graphics (SGI), die für ihre Workstation eine spezielle Grafikkarte entwickelt hatten.

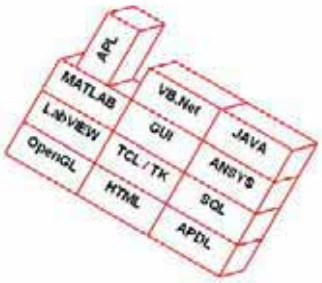
ANZEIGE

Produktentwicklung
Automatisierungstechnik

Software-Erstellung auf Basis matrizenorientierter Methoden

Graphische Datenverarbeitung

Client / Server – Modelle



Dipl.-Ing. Helmut Mumbauer
Stormweg 12
41564 Kaarst

www.APL-Software.de
Helmut.Mumbauer@T-Online.de

Tel/Fax: 02131 / 6 49 74 Mobil: 0173 / 210 3000

Kontakt: Helmut Mumbauer

APL4CVS Versionsmanagement mit APL + CVS

- Projektarbeit mit APL
- Große Freiheit - große Gefahren
- Arbeiten in Entwicklerteams
- Updatemechanismen und Versionsmanagement
- Risiken und Fehlerquellen
- Kundenanforderung an zukünftige APL-Projekte
- Eigenes Versionsmanagement entwickeln oder Vorhandene nutzen?

Arbeiten in Entwicklerteams:

Für ein erfolgreiches Arbeiten in Projektteams ist es erfoderlich das Standards definert werden. Diese Standards beziehen sich auf:

- die Dokumentation der APL-Objekte
- die Namenskonventionen der APL-Objekte
- die Struktur eines Workspaces
- der Umgang mit gemeinschaftlich genutzten Objekten

Im nächsten Schritt müssen Mechanismen definiert werden, wie das Zusamenführen verschiedener Entwicklungsworkspaces vollzogen wird.

Updatemechanismen und Versionsmanagement?

Eine APL-Anwendung besteht aus einem APL-Workspace. Ein Update erfolgt somit über:

- einen neuen, zu verteilenden Workspace
- das Austauschen und Einpflegen von APL-Objekten über APL-Files
- das ‚Zusammensetzen' einer APL-Anwendung über Datenbankobjekte

Je nach gewählter Update-Methode kann natürlich ein entsprechendes Versionsmanagement konstruiert werden.

Risiken und Fehlerquellen

- sehr großer Verwaltungsaufwand
- redundante APL-Objekte entwickeln sich auseinander
- hohe Einarbeitungszeit bei neuen Projektmitarbeitern

Kundenanforderung an zukünftige APL-Projekte

- Fehlendes Versionsmanagement ist KO-Kriterium für APL-Projekte
- Moderne Entwicklungswerkzeuge verfügen über Versions- und UpdateManagement

Eigenes Versionsmanagement entwickeln oder Vorhandene nutzen?

- Seit 1970 werden Versionsmanagement Systeme entwickelt
- Erfahrung vorhandener Systeme nutzen
- APL an die Systeme anpassen!

Kontakt: Bernd Stolle,
a.k.e Consult GmbH, bernd.stolle@ake-consult.de

Beispiele für APL + CVS

Schritt 1: Import

APL
- Neuen Workspace erstellen
- Einige erste Funktionen als Basis für das neue Projekt anlegen
- Import ins CVS Repository durchführen

WinCVS
- Das neue Projekt als Modul in das CVS Repository importieren

Schritt 2: Checkout

WinCVS
- Der User erzeugt ein eigenes Arbeitsverzeichnis
- Das Projekt zur Bearbeitung wird aus dem CVS-Repository auf die lokale Festplatte kopiert

APL
- Das Modul ins APL laden
- Der Workspace wird bearbeitet
- Die lokale Kopie des Workspaces wird gespeichert

Schritt 3: Commit

WinCVS
- Der User erzeugt ein eigenes Arbeitsverzeichnis
- Das Projekt zur Bearbeitung wird aus dem CVS-Repository auf die lokale Festplatte kopiert

APL
- Das Modul ins APL laden
- Der Workspace wird bearbeitet
- Die lokale Kopie des Workspaces wird gespeichert

WinCVS
- Der geänderte Workspace wird in das CVS-Repository hochgeladen

Schritt 4: Update

WinCVS
- Der User will seinen geänderten Workspace in das VS-Repository hochladen
- Die lokale Version muss zuerst aktualisiert werden

APL
- Der Workspace wird geladen
- Die von User2 geänderten Funktionen stehen zur Verfügung

Schritt 5: Konfliktfall

APL
- User2 hat die gleiche Funktions geändert wie User1

WinCVS
- Versuchen, die Änderungen in das CVS-Repository hochzuladen
- Versuchen, die lokale Kopie zu aktualisieren
- Wie kann der Konflikt gelöst werden?

“Schöne APL-Programme“ – mal anders

Einführung

Mit Dyalog.net bzw. Version 10 führte Dyalog die Möglichkeit ein, einen Workspace als .EXE zu „kompilieren“ (bzw. eine .EXE zu erstellen, in die der Workspace eingebunden ist und mit Hilfe des Interpreters in einer .DLL ausgeführt wird.) Die entstehenden .EXE-Dateien sind klein und handlich, benötigen aber leider die >3MB große Interpreter-DLL, aber dennoch ist dieser Weg, eine Applikation zu „packen“, komfortabel und erleichtert es, Anwendungen weiterzugeben (zum Programmstart werden keine Verknüpfungen zum RT benötigt etc.).

Die einzige Schwierigkeit, die ich mit den EXEs hatte, war, dass Dyalog (fast) keinen Zugriff auf die erweiterten Dateieigenschaften der .EXE gab (Versionsinformation, Autor etc.), und so suchte ich nach Wegen, diese Daten zu ändern. Der Artikel beschreibt meine Ergebnisse[1].

Die Zutaten

Dyalogs .EXE-Dateien werden im “Win32PE-Format” erzeugt. “PE” bedeutet dabei “Portable Executable” und ist das Standard-Format für Applikationen in der Win32-Welt (und mit einigen Modifikationen auch unter Win64). Mehr Informationen über dieses Format sind im Web zu finden, eine umfangreiche Darstellung z.Bsp. unter [1]. Entscheidend ist, dass dies ein standardisiertes Format ist und wir somit standardisierte Werkzeuge zum Bearbeiten der Dateien nutzen können, es gibt keine „verborgenen APL-Spezialitäten“, die dem entgegenstünden.

Im WWW sind verschiedenste Werkzeuge zum Bearbeiten von .EXE-Dateien zu finden. Das Tool, mit dem ich gearbeitet habe, nennt sich Resource Hacker [2] und hat den großen Vorteil, als Freeware verfügbar zu sein.

Mit Resource Hacker ist es möglich, die Versionsinformationen einer .EXE-Datei zu bearbeiten und Icons hinzuzufügen – interaktiv. Das Programm hat auch einen Befehlszeilenmodus und unterstützt Skripting, doch in diesem Modus werden bereits kompilierte Resourcen benötigt, die dann die EXE-Datei eingefügt werden.

Schon wieder kompilieren?!

Um also den ganzen Vorgang zu automatisieren, ist es nötig, eine Resourcen-Datei (.RC) zu erzeugen, mit einem Resource-Compiler zu kompilieren und dann mit Resource Hacker die Resourcen der .EXE zu aktualisieren.

Erfreulicherweise ist das alles nicht so schwierig, wie es klingen mag. Zu Visual Studio gehört auch ein Resource-Compiler (Common\Tools\Bin\RC.EXE) and wer kein VS zur Hand hat, kann auch GoRC [3] als Freeware verwenden[2].

Alles zusammen

Ich verwende eine Funktion `∆Build∇`, um Versionsnummern zu setzen, den Workspace zu sichern, eine passende .RC-Datei zu schreiben, diese zu kompilieren, den Workspace zu compilieren und schliesslich mit Resource Hacker die Resourcen der .EXE zu aktualisieren.

```
[0] Build;this;nu;txt;cmd
[1] this←{(⍵∧\⍵≠'\')/⍵}⎕WSID
[2] :If save
[3]    write_rc
[4]    cmd←'c:\dyalog10\dotnet\aplc.exe /v'
[5]    cmd,←'/t:nativeexe ',this,'.dws'
[6]    cmd,←(FileX this,'.ico')/' /icon:',this,'.ico'
[7]    ⎕←⎕CMD ⎕←cmd
[8]   ⍝cmd←'"C:\Programme\Microsoft Visual Studio .NET'
[9]   ⍝cmd,←'2003\Common7\Tools\Bin\RC.Exe" /r /v '
[10]  ⍝cmd,←,this,'.rc'
[11]  ⍝ ⎕←⎕CMD ⎕←cmd
[12]  ⎕←⎕CMD ⎕←'c:\tools\gorc\gorc /r ',this,'.rc'
[13]   :If FileX this,'.rh_script'
[14]      cmd←'c:\tools\reshack\reshacker.exe'
[15]      cmd,←' -script ',this,'.rh_script'
[16]   :Else
[17]      cmd←'c:\tools\reshack\reshacker.exe'
[18]      cmd,←'-addoverwrite "', ⎕WSID
[19]      cmd,←'.exe","',⎕WSID,'.exe","'
[20]      cmd,←,⎕WSID,'.res",versioninfo,1,1033'
[21]    :EndIf
[22]    ⎕←⎕CMD ⎕←cmd
[23]    nu←'c:\tools\reshack\reshacker.log'⎕NTIE 0
[24]    txt←⎕NREAD nu,82,⎕NSIZE nu ⋄ ⎕NUNTIE nu
[25] ⍝ Zeilenumbruch bei Ausgabe Ergebnis
[26]    {{}⎕←1↓⍵}¨{(⍵=⎕TC[2])⊂⍵}⎕TC[2],txt~⎕TC[3]
[27] :EndIf
```

Fussnoten:

1 Ergänzung: Version 10.1 bietet erweiterte Möglichkeiten, diese Dateieigenschaften zu setzen, so dass ich zunächst dachte, der Artikel sei nun überflüssig. Das ist aber nicht der Fall, denn die hier beschriebenen Möglichkeiten gehen über die von Dyalog angebotenen Funktionen hinaus und bieten Optionen, die in Dyalogs Werkzeugen (noch?) nicht enthalten sind: es ist möglich, Build-Prozesse zu automatisieren, da die Eigenschaften auch unter Programmkontrolle gesetzt werden können und zudem zusätzliche Resourcen wie etwa Icons, in die .EXE-Dateien eingefügt werden können.

2 Ich hatte Schwierigkeiten, als ich GoRC verwendete: die EXE zeigte auch nach Einfügen der kompilierten Resourcen noch alte Versionsinformationen an. Mit tatkräftiger Unterstützung des GoRC-Autors konnte das Problem schliesslich geklärt werden: in der von ResourceHacker erzeugten Datei fehlten die Sprachinformationen und in diesem Fall verwendete GoRC die Einstellungen des Betriebssystems. Da ich aber auf einem deutschen Windows eine EXE-Datei mit englischer Programminformation erzeugen wollte, ging das schief. Mit Microsofts RC war das erstaunlicherweise kein Problem, weil dort als Default Englisch hardcodiert ist, was in diesem Fall ja auch korrekt war.
Abhilfe: die erste Zeile der .RC-Datei sollte Sprachinformationen enthalten, etwa „`LANGUAGE LANG_ENGLISH,SUBLANG_ENGLISH_US`“

Die nicht aufgeführte Funktion `save` aktualisiert die Versionsinformation und `⎕SAVEd` den Workspace. Die nachfolgende Funktion `write_rc` benutzt eine Variable als Vorlage und aktualisiert die Programminformationen darin (indem versucht wird, alle in % eingeschlossenen Strings zu aktivieren) und schreibt das Ergebnis in eine Datei:

```
write_rc;txt;z;apl;file;nu
txt←myΔrc  ⍝ Vorlage
txt←txt,¨⊂⎕TC[3 2]
txt←qENLIST txt
⍝ suche nach APL-Anweisungen, in '%' eingeschlossen
z←('%'⍷txt)∧~{⍵∨0,¯1↓⍵}'%%'⍷txt
:While ∨/z
   apl←1↓(1=+\z)/txt
   :Trap 0 ⋄ apl←⍎⍕apl ⋄ :EndTrap  ⍝ aktivieren
   txt←((¯1+z⍳1)↑txt),apl,(1⍳⍨¨2=+\z)↓txt ⍝ersetze durch res
   z←('%'⍷txt)∧~{⍵∨0,¯1↓⍵}'%%'⍷txt
:EndWhile
z←'%%'⍷txt ⋄ txt←(~z)/txt ⍝ doppelte '%%' umwandeln in '%'
⍝ Datei schreiben
:Trap 0  ⍝ löschen, falls vorhanden
   nu←(file←⎕WSID,'.rc')⎕NTIE 0 ⋄ file ⎕NERASE nu
:EndTrap
nu←file ⎕NCREATE 0
txt ⎕NAPPEND nu
⎕NUNTIE nu
```

Die Variable `myΔrc` schließlich hat folgenden Inhalt:

```
LANGUAGE LANG_ENGLISH,SUBLANG_ENGLISH_US
1 VERSIONINFO
FILEVERSION %GetVersion 1%,%GetVersion 2%,%GetVersion
3%,%GetVersion 4%
PRODUCTVERSION %GetVersion 1%,%GetVersion 2%,%GetVersion
3%,%GetVersion 4%
FILEOS 0x4
FILETYPE 0x2
{
BLOCK "StringFileInfo"
{
    BLOCK "040904B0"
    {
        VALUE "CompanyName", "Ich selbst"
        VALUE "FileDescription", "Tolle Anwendung"
        VALUE "FileVersion", "%GetVersion 1%,%GetVersion
2%,%GetVersion 3%,%GetVersion 4%"
        VALUE "InternalName", "TolleAnwendung"
        VALUE "LegalCopyright", "Copyright (C) 2005 Me"
        VALUE "OriginalFilename", "TolleAnwendung.EXE"
        VALUE "ProductName", "TolleAnwendung"
        VALUE "ProductVersion", "%GetVersion
1%,%GetVersion 2%,%GetVersion 3%,%GetVersion 4%"
        VALUE "Website", "http://www.TolleWebsite.de"
        VALUE "Support", "support@TolleWebsite.de"
    }
}
BLOCK "VarFileInfo"
   {
    VALUE "Translation", 0x0409 0x04B0
   }
}
```

Meines Wissens nach gibt es keine verbindlichen Regeln für die Inhalte des StringFileInfo-Bereichs, man kann also eigene Parameter dort anlegen. Die Funktion `GetVersion` gibt die Versionsnummer (oder Teile davon, je nach Argument) zurück. Auf meiner Website [4] steht ein kompletter Beispielworkspace zum Download zur Verfügung. Darin ist auch die `save`-Funktion enthalten und andere Kleinigkeiten, auf die aus Platzgründen im Artikel verzichtet wurde.

Literaturhinweise:
[1] http://msdn.microsoft.com/msdnmag/issues/02/02/PE/default.aspx
[2] http://www.users.on.net/johnson/resourcehacker/
[3] http://www.godevtool.com/
[4] http://mbaas.de

Kontakt:
Michael Baas
Email: michael@mbaas.de

Die kleine Maus hat keinen Schwanz

Die kleine Maus saß vergnügt im Regal. Rechts und links von ihr saßen soweit sie schauen konnte Mäuse. Sie sah graue Mäuse und schwarze, scheckige und ganz einfarbige. Alle warteten sie darauf, dass der Laden öffnen würde. Heute Morgen war die kleine Maus zwischen all die anderen ins Regal gesetzt worden. Zuerst hatte sie sich nur ganz vorsichtig umgeschaut. Neben ihr saß eine totschicke silberfarbige Maus. Sie war wesentlich größer als sie selbst und passte kaum auf das Regalbrett, auf dem sie hockten. Die kleine Maus selbst fand ihre leicht blaue Farbe sehr schön. Jedenfalls bekam sie gute Laune, als sie an sich heruntersah. Sie blickte froh und erwartungsvoll in den Tag.

Der Laden öffnete. Ein Junge steuerte zielsicher auf ihr Regal zu, eine junge Frau begeistert hinter sich her ziehend. „Da ist sie!“ rief er aus, „Die möchte ich gerne haben. Mama, bitte!“

„Das ist ja toll!“ dachte die kleine Maus. „Ich wurde doch gerade erst in das Regal gesetzt und schon interessiert sich jemand für mich.“ Sie saß ganz still und blickte Mutter und Sohn hoffnungsvoll an.

Die Mutter holte erst mal Luft, verschnaufte etwas, betrachtete interessiert die kleine Maus und blieb still.

Die kleine Maus hatte sich von der Begeisterung des Jungen anstecken lassen und schaute jetzt wie er voller Erwartung auf die junge Frau. Irgendwann musste diese ja reagieren. Es dauerte immernoch etwas.

„Nein!“ antwortete sie, „Schau genau hin! Diese Maus hat keinen Schwanz. Du weisst doch, Mäuse ohne Schwanz können nicht gut springen. Das hast du jeden Tag schon beobachten können.“ Die Mutter lachte ihren Sohn freundlich an, schüttelte sanft den Kopf und zog ihn bestimmt weiter.

Traurig schaute die kleine Maus dem unwillig folgenden und enttäuschten Jungen nach. „Schade! Aber vielleicht geht das auch nicht so schnell. Gleich am ersten Tag, das wäre wohl übertrieben schön gewesen“, dachte sie.

Der Junge blickte sich immer wieder sehnsüchtig nach ihr um, schien aber hilflos dem lachenden Zug seiner Mutter nichts entgegen zu setzen zu haben.

„Die Mutter hat etwas vom Springen erzählt. Springen? Was ist das?“, fragte sie sich nun. Die kleine Maus war doch noch nie gesprungen. Verstohlen beobachtete sie die anderen. Sie beobachtete sie sehr genau, den ganzen Vormittag hindurch. Keine der anderen Mäuse tat ihr den Gefallen und sprang auch nur ein winziges Stück. Obwohl alle, jedenfalls soweit sie es sehen konnte, einen langen Schwanz hatten. Sie entschied sich, dass mit dem Springen selbst zu versuchen, auch wenn sie nicht über einen Schwanz verfügte.

Die kleine Maus duckte sich so tief sie konnte, noch etwas tiefer. Dann spannte sie alle Gedanken an und machte einen mächtig gewaltigen Satz - in Gedanken! Wieder beobachtete sie genau ihre Umgebung. Aber enttäuscht musste sie sich eingestehen, dass niemand ihren virtuellen Sprung wahrgenommen hatte, oder besser, einen Sprung überhaupt hätte wahrnehmen können. Sie hatte sich nämlich nicht einen Millimeter fortbewegt, in keiner Richtung.

Für den Rest des Tages war sie damit beschäftigt entweder ihre Mäusekumpane zu beobachten, sie bewegten sich eindeutig nicht, oder selbst springen zu üben, sie persönlich konnte es bisher einfach nicht.

Die Nacht war lang. Die kleine Maus konnte nicht schlafen. Im Dunkeln machte sie weitere Versuche zu springen. Insgeheim hoffte sie, am Morgen würde sie ihre nächtlichen Bewegungserfolge sehen können. Sie hatte sogar ein wenig Angst, vielleicht vom Regalbrett zu fallen. Aber am Morgen musste sie resigniert feststellen, dass keine Maus seine Position geändert hatte, auch nicht die Mäusekollegen, die über einen Schwanz verfügten. Der Laden öffnete und der Junge erschien heute ohne seine Mutter. Voller Sehnsucht betrachtete er die kleine Maus eine kurze Unendlichkeit lang. Ganz zärtlich legte er seine Hand auf sie. Das fühlte sich richtig gut an. Dann rannte er plötzlich davon. Die kleine Maus schaute umgekehrt jetzt ihm sehnsüchtig nach. Die sanfte Berührung, in die sie sich hatte hineinkuscheln können, stachelte ihre Bemühungen erneut an. Auch wenn hier niemand sprang, auch die Mäuse mit einem Schwanz nicht, die kleine Maus wollte springen.

Sie wollte ganz hoch springen lernen, vielleicht würde die Mutter des Jungen doch zu überzeugen sein und sie durfte zu ihm. In der Zwischenzeit konnte sie beobachten das manchmal die eine oder andere Maus mit langem Schwanz verkauft wurde. Aber nie eine ihrer Kumpane direkt vom Regalbrett, sondern der Verkäufer holte stets eine Maus mit Kiste aus einem der unteren Regalbretter. „Seltsam!“

Der Junge kam jeden Tag, kurz nach der Ladenöffnung. Die Sehnsucht seiner Augen und seine sanften Hände begleiteten sie von nun an durch alle Tage. Aber die kleine Maus lernte nicht springen. Sie dachte sich so viele Möglichkeiten wie ihr nur einfielen aus, aber sie konnte einfach nicht springen. Doch sie begriff, was Sehnsucht ist.

Dann kam ein Tag, der alles veränderte. Der Junge erschien wie üblich kurz nach der Ladenöffnung. Diesmal blieb er etwas von ihr entfernt stehen. Er wirkte heute anders. Seinen Augen strahlten selbstsicher. Er stand etwas breitbeinig und kramte energisch in seiner Hosentasche herum. Nach einiger Mühe beförderte er einen Haufen Münzen zu Tage, versuchte sie in einer seiner Hände zu bergen und lief am Regal entlang auf die Verkaufstheke zu. Mit einem der kleinen Maus gut bekannten freundlichen Verkäufer kam er zurück.

„Diese Maus dort möchte ich erstehen.“ Die Stimme des Jungen war heute ganz sicher und bestimmt und er zeigt wirklich genau auf sie.

„Gut“, sagte der Verkäufer. Er griff wie so oft von ihr beobachtet unter ihr ins Regal und hielt dem Jungen eine Kiste hin. „Hier - eine ....“

„Nein!“ empörte sich der Junge sofort, „Ich möchte nur diese dort und auf gar keinen Fall irgend eine andere. Genau sie ist es, die ich jeden Tag gesehen habe. Ihre Farbe hat einen kleinen Fleck auf der rechten Seite. An ihn bin ich gewöhnt, ihn habe ich jeden Tag berührt. Ich nehme nur diese eine Maus und keine andere!“ Der Junge klang so bestimmt, dass der Verkäufer kurz schmunzelnd, lächelnd die Achseln zuckte. Er nahm die kleine Maus direkt vom Regal, dazu eine leere Schachtel vom Brett unter ihr und legte sie hinein. Es wurde dunkel, als er die Schachtel schloß. Die kleine Maus konnte an der Bewegung erahnen, dass sie getragen wurde. Ab und zu schien es ihr, als ob sie abgelegt wurde. Dann nahmen die Bewegungen wieder zu. Je länger es dauerte, desdo aufgeregter machte sie die Ungewissheit.

War es wirklich wahr, dass sie jetzt zu dem Jungen gehören sollte? Was erwartete sie wohl, wenn die Schachtel wieder geöffnet würde. Wonach hatte sie sich gesehnt? Die Wiederkehr der täglichen Begegnungen mit dem Jungen im Laden, seine sanften Hände, hatten ihrer Sehnsucht eine genaue Vorstellung zugesellt. Jetzt würde auf jeden Fall alles anders sein. Sie konnte sich im Moment nicht vorstellen, wonach sie sich gesehnt hatte, was hätte folgen sollen.

Nach langem Weg wurde es zuerst einmal bewegungslos still. Dann öffnete sich die Schachtel, es schien Licht herein und die kleine Maus wurde sanft von den ihr bekannten Händen auf ein schönes Bild gesetzt. Sie schaute sich um. Vor ihr stand ein großer Bildschirm. Neben ihr etwas langes mit Tasten konnte sie nicht begreifen. Aber sie saß auf einem wunderschönen Sanddünenbild direkt vor dem Bildschirm auf einem weißen Tisch. In einem Regal hinten an der Wand waren mehre Käfige zu sehen, in denen etwas pelziges hin und her huschte. Sie schaute länger und genauer hin, das pelzige Etwas konnte ja springen! Ab und zu sprang es an der Käfigwand nach oben. „Hoch Interessant!“

Die Mutter des Jungen kam herein. „Na, hast du endlich genug Mäuse für deine ganz besondere Computermaus ohne Schwanz zusammengespart? Jetzt kannst du ihr ja das Springen beibringen!“ lachte sie.

„Das lernt sie ganz, ganz schnell“, antwortete der Junge. „Schau doch zu!“

Er tippte auf der Tastatur, „Aha“, der Bildschirm veränderte seine Farben. Der Junge legte seine Hand erneut über die kleine Maus, ganz leicht. Seine Finger drückten eine ihrer Tasten und dann bewegte er sie mit winzigen, leicht gleitenden Schritten über das Sanddünenbild unter ihr. Auf dem Bildschirm sprang ein kleiner Zeiger fröhlich im Zickzack hin und her.

„Siehst du wie sie auf dem Pad springen kann? Weiter und vor allem schneller als alle deine Mäuse dort drüben im Käfig, und das, obwohl meine Maus keinen Schwanz hat.“ Der Junge lachte seine Mutter glücklich an. „Eine lichtgesteuerte Funkmaus! Sie braucht keinen altmodischen Schwanz. Technik - nicht Biologie!“

„Naja, bis sich deine Technikmaus zu dieser speziellen Art entwickelt hatte, musste sie wie meine lebendigen Mäuse auch eine Evolution durchmachen, Technikrevolution, sonst könnte sie nicht so auf ihrem Lichtstrahl über ihren Sand flitzen und die Sprünge auf den Bildschirm funken. Und du hättest nicht soo lange warten müssen auf deine, kleine, teure Supermaus!“

„Miau, Supermausi! Los geht`s!“ lachte er und sie durfte erneut über seinen Bildschirm huschen, seinen sanften Händen immer so nahe.

Barbara Pütz, vindur13@gmx.de

Weltweit erstes Computermodell der Haut entwickelt

Experimente bestätigen erste Ergebnisse

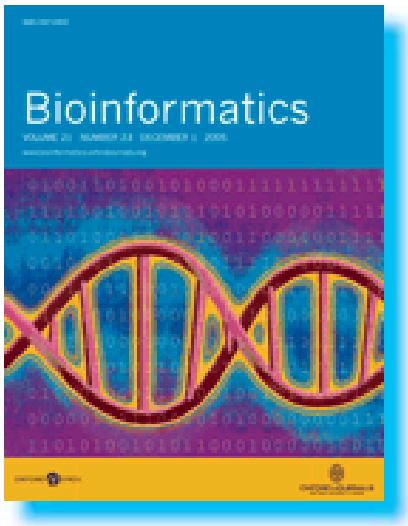
Hamburg. Wissenschaftler aus Hamburg haben das weltweit erste Computermodell der menschlichen Haut entwickelt. Die Arbeitsergebnisse des Bioinformatikers Dr.-Ing. Niels Grabe und des Dermatologen Prof. Dr. Karsten Neuber (Universitätsklinikum Hamburg-Eppendorf) wurden jetzt in der internationalen renommierten Fachzeitschrift „Bioinformatics“ veröffentlicht. Das Computermodell dient dem besseren Verständnis der Hautentstehung und bietet damit die Chance, neue Therapieansätze zu entwickeln.

Für experimentelle Untersuchungen von Organfunktionen und Stoffwechsel des Menschen stehen bisher in der Regel nur Modelle zur Verfügung, die die komplexen tatsächlichen Vorgänge im Körper sehr stark vereinfachen (Zellkulturen) oder vom menschlichen Organismus mehr oder weniger weit entfernt sind (Tiermodelle). Zudem gelingt es häufig nicht, Detailerkenntnisse zu einem entwicklungsfähigen Ganzen zusammenzufügen. Die so genannte Systembiologie, die Computer- und Labortechnologie vereint, bietet erstmals die Möglichkeit, die Auswirkungen komplexer molekularer Netzwerke auf ein gesamtes Gewebe zu simulieren.

In ihrem Forschungsprojekt, das an der Klinik für Dermatologie und Venerologie des Universitätsklinikums Hamburg-Eppendorf und beim Zentrum für Bioinformatik der Universität Hamburg angesiedelt war, fütterten die Wissenschaftler Karsten Neuber und Niels Grabe den Computer mit bereits bekannten Daten zur Entstehung und Erneuerung der menschlichen Epidermis, des mehrschichtigen äußeren Teils der Haut. Für jede einzelne daran beteiligte Zellart wurde dabei von Dr. Grabe ein eigenes kleines Programm geschrieben, das die grundlegenden Fertigkeiten der Zelle berücksichtigt und selbstständig abläuft. Das auf diese Weise entstandene zweidimensionale Modell simuliert, wie sich einzelne Stammzellen unter dem Einfluss im Computer nachempfundener molekularer Mechanismen kontrolliert teilen und so langsam die einzelnen Schichten der Haut entstehen.

Erste Ergebnisse des Computermodells sind mittlerweile durch experimentelle Untersuchungen bestätigt worden. Die Forscher erhoffen sich von dem Computermodell insbesondere neue Perspektiven zur Therapie der stark zunehmenden Hautkrebserkrankungen. Bei dem Modell handelt es sich jedoch, so Professor Neuber, „erst um einen Prototypen, der nur dann erfolgreich sein kann, wenn er als interdisziplinäre Kooperation mit Forschergruppen aus den verschiedenen relevanten Gebieten erweitert wird.“

Dr.-Ing. Niels Grabe, entwickelte gemeinsam mit Prof. Dr. Karsten Neuber ein weltweit erstes Computermodell der Haut. (Foto: Zentrum für Bioinformatik, Universität Hamburg)



Literatur:

Grabe, Niels; Neuber, Karsten: A multicellular systems biology model predicts epidermal morphology, kinetics and Ca2+ flow. In: Bioinformatics, Sep 2005; 21: 3541 - 3547.

Kontakt: Prof. Dr. Karsten Neuber, Klinik für Dermatologie und Venerologie, Universitätsklinikum Hamburg-Eppendorf, Martinistr. 52, D-20246 Hamburg, Tel. 040.42803-2642, Internet: www.uke.uni-hamburg.de.

Dr.-Ing. Niels Grabe, Universitätsklinikum Heidelberg, Institut für Medizinische Biometrie und Informatik, Abteilung Medizinische Informatik, Im Neuenheimer Feld 400, D-69120 Heidelberg, Tel. 06221.56-7483, Fax: -56-4997, eMail: Niels.Grabe@med.uni-heidelberg.de, Internet: www.klinikum.uni-heidelberg.de.

Zentrum für Bioinformatik, Universität Hamburg, Bundesstraße 43, D-20146 Hamburg, Tel. 040.42838 7373, Fax -42838 7352, eMail: info@zbh.uni-hamburg.de, Internet: www.zbh.uni-hamburg.de/index.php.

Der digitale Organismus

Karlsruher Forscher hauchen Computern Intelligenz ein

Karlsruhe. Computer sind vielseitig. Sie führen Verkaufsverhandlungen, organisieren Auktionen oder übernehmen die Steuerung von Maschinen. Trotzdem ist ihre Handlungsweise eingeschränkt. Eine neue Aufgabe erkennen und flexibel darauf reagieren können sie bisher nicht. Computer leisten Auftragsarbeit, für die sie programmiert sind. Ändert sich die Aufgabe, muss die Programmierung geändert werden. An der Universität Karlsruhe haucht nun eine Forschergruppe Rechnern „Intelligenz ein“: Sie orientiert sich an Mechanismen des menschlichen Körpers.

Organic Computing heißt diese Wissenschaftsrichtung, die gezielt nach Parallelen in der Natur sucht, um bestimmte Aufgaben zu lösen. „Wir möchten, dass sich ein Rechner selbstständig an verschiedene Situationen anpasst“, beschreibt Professor Dr. Jürgen Becker. Digitalen Organismus - diesen Begriff hat der Sprecher des Forschungsvorhabens und Leiter des Instituts für Technik der Informationsverarbeitung für das Untersuchungsobjekt geprägt, an dem er gemeinsam mit fünf weiteren Wissenschaftlern aus den Bereichen Informatik, Elektrotechnik und Biologie arbeitet. Ihr Untersuchungsobjekt: Ein Industrie-Roboter. Er soll später in der Lage sein, zu entscheiden, welche Aufgabe im Moment am sinnvollsten ist und diese in Angriff nehmen. Fehlt beispielsweise ein Zwischenprodukt, entscheidet der Roboter, wann er mit dessen Fertigung anfängt. Becker: „So wird die Effizienz des Fertigungsprozesses gesteigert.“

Die Forschergruppe plant, den digitalen Organismus an ausgewählte Mechanismen im menschlichen Organismus anzulehnen: Mit Zellen, Organen und Gehirn. Die kleinsten Einheiten - die Zellen - bilden einzelne Prozessorelemente. Sie müssen sich gegenseitig verständigen und andere Zellen bei Bedarf ersetzen können - analog einem Zellverband bei Säugetieren. Weiter müssen diese „Zellen“ rekonfigurierbar sein. Becker: „Nur dadurch kann der digitale Organismus flexibel und schnell genug auf Anforderungen von außen reagieren.“

Die zweite Hierarchieebene gestalten die Karlsruher Wissenschaftler in Anlehnung an die Organe des menschlichen Körpers. Einzelne Zellen der ersten Ebene schließen sich zu „Organ-Verbänden“ zusammen und übernehmen eine aktuelle Aufgabe. Den „Befehl“ für den Zusammenschluss erhalten sie über „virtuelle Botenstoffe“ in Form elektronischer Botschaften. Über die Organe wacht auf oberster Ebene das Gehirn. Es erkennt die Anforderungen der Umwelt sowie den inneren Systemzustand und gibt dementsprechend die Befehle an die darunter liegenden Ebenen weiter.

Das Forschungsvorhaben ist ein Teilprojekt des Schwerpunktprogramms Organic Computing, das die Deutsche Forschungsgemeinschaft (DFG) im vergangenen Jahr eingerichtet hat. Diesen Sommer hat die DFG aus 59 eingereichten Projektvorschlägen 18 Projekte bewilligt. Drei davon sind an der Universität Karlsruhe angesiedelt. Sprecher des Schwerpunktprogramms ist Professor Dr. Hartmut Schmeck vom Institut für Angewandte Informatik und Formale Beschreibungsverfahren der Universität Karlsruhe.

Kontakt: Professor Dr. Jürgen Becker, Institut für Technik der Informationsverarbeitung, Engesserstraße 5, D-76131 Karlsruhe, Tel. 0721.608-25 02, Fax -607 438, eMail: becker@itiv.

Natürlichsprachige Suchfunktion

Computerlinguisten vereinfachen Suche im Brockhaus

Saarbrücken. In der neuen Auflage der Brockhaus Enzyklopädie kann man erstmals mit Hilfe einer natürlichsprachigen Suchfunktion recherchieren. In dem umfangreichen Lexikon kommen neue statistische und linguistische Extraktionsmethoden und Visualisierungstechniken zum Einsatz, die in einem vom Bundesministerium für Bildung und Forschung geförderten Projekt entwickelt wurden. Die neue Software, die Computerlinguisten am Saarbrücker Institut für angewandte Informationsforschung (IAI) programmiert haben, kann mit Fragen wie „Was ist der Zusammenhang zwischen Übergewicht und Herzinfarkt?“ oder „Warum ist der Himmel blau?“ umgehen und den Leser zu einem Artikel führen, der Antworten auf diese Fragen gibt. „Diese Art der Suche ist bisher einzigartig und besonders nützlich, wenn der gesuchte Begriff unbekannt ist“, so Bernd Kreissig, Geschäftsführer der Brockhaus Duden Neue Medien GmbH. Der neue Brockhaus, der kürzlich auf der Frankfurter Buchmesse präsentiert wurde, enthält 300.000 Stichwörter auf 24.500 Seiten in 30 Bänden. Wer im Bücherregal nicht so viel Platz hat, kann den kompletten Textinhalt jetzt auf einem USB-Stick erwerben.

Kontakt: Prof. Dr. Johann Haller, Lehrstuhl für maschinelle Übersetzung, Universität des Saarlandes, Tel. 0681.302-4484, eMail: hans@iai.uni-sb.de, Internet: www.iai-sb.de.

Friederike Meyer zu Tittingdorf, Kompetenzzentrum Informatik der Universität des Saarlandes, Tel. 0681.302-58099, Internet: www.informatik-saarland.de.

Netzwerkbetreiber als Innovationsbremse

TU-Wissenschaftlerin sieht Internet am Scheideweg

Berlin. Das Internet ist mit Entwicklungen wie E-Mail oder der Internet-Telefonie ein Tummelplatz für Innovationen. Doch der bisherige Garant des Fortschritts, die Offenheit der Übertragungsnetze, droht durch die Restriktionen von Netzbetreibern ins Wanken zu geraten, warnt die Wissenschaftlerin Dr.-Ing. Barbara van Schewick vom Fachgebiet Telekommunikatiosnetze an der Technischen Universität (TU) Berlin. Einschränkungen der Offenheit haben einschneidende wirtschaftliche Konsequenzen, sie behindern die Entstehung neuer Innovationen, wie van Schewick in ihrer Dissertation „Architecture and Innovation: The Role of the End-to-End Arguments in the Original Internet“ festgestellt hat. Für ihre Arbeit wurde die Informatikerin und Juristin mit dem Wissenschaftspreis der Deutschen Stiftung für Recht und Informatik (DSRI) ausgezeichnet. Der Preis ist mit 2.000 Euro dotiert.

Die TU-Wissenschaftlerin konnte erstmals die Theorie belegen, dass die Vielzahl der Innovationen im Internet kein Zufall ist, sondern durch die ursprüngliche Architektur des Internets ermöglicht wurde. Die Internetarchitektur beruht auf einem Design-Prinzip, das End-to-End-Argument genannt wird. Bei diesem Prinzip laufen die Anwendungen wie etwa ein Internetbrowser auf dem heimischen PC. Das Netzwerk sorgt lediglich für die Übertragung der Daten; es kann nicht erkennen, welche Anwendungen gerade laufen.

Inzwischen wird jedoch von der ursprünglichen Architektur abgewichen. So benutzen Netzwerkbetreiber verstärkt Technologie, die es ihnen ermöglicht, zwischen Anwendungen zu unterscheiden und ihre Ausführung nach eigenem Gutdünken zu beeinflussen. Bestimmte Anwendungen könnten ganz ausgeschlossen werden oder ein Netzwerkbetreiber verlangsamt die Anwendung eines Konkurrenten, ohne dass die Kunden davon etwas ahnen. In den USA, so die TU-Wissenschaftlerin, sei dies mit der Software eines Internet-Telefonie-Anbieters passiert.

Als Beispiele für Einschränkungen der Netze, die den Fortschritt behindern können, nennt die TU-Wissenschaftlerin Überlegungen des Mobilfunkunternehmens Vodafone, Internet-Telefonie-Signale im UMTS-Netz zu sperren. Kontraproduktiv sei auch die Drohung der Deutschen Telekom AG, nur dann neue Hochgeschwindigkeitsnetze mit einer Übertragungsrate von 50 Megabit pro Sekunde zu bauen, wenn der Konzern die Netze entgegen der bisherigen Regelungen für einige Jahre alleine nutzen darf.

In der Konsequenz, so van Schewick, würde die Kontrolle des Netzes durch die Betreiber dazu führen, dass einige Anwendungen vom Markt gedrängt werden oder nicht auf den Markt kommen. Für Netzwerkbetreiber könne das durchaus lukrativ sein, doch für unabhängige Entwickler würde es schwerer, leicht und billig Innovationen gewinnbringend zu vermarkten. Damit würde folglich der Anreiz sinken, an innovativen Anwendungen und Produkten zu forschen.

Neuerungen, die auch den Netzwerkbetreibern neue oder zusätzliche Einnahmemöglichkeiten eröffnen würden, würden dann ausbleiben, so van Schewick. Denn planbar seien Erfolge nicht, wie der Siegeszug von Beispielen wie E-Mail, Word Wide Web oder Musiktauschbörsen zeige. „Die Offenheit der Netze ist daher die Grundvoraussetzung, um weiterhin eine Vielfalt von Innovationen im Internet zu ermöglichen. Das Internet steht nun am Scheideweg. Wettbewerb alleine ist nicht die Lösung. Es geht darum, welche Architektur sich in Zukunft durchsetzen wird“, zieht Barbara van Schewick Bilanz.

Im Mutterland des Internets, den USA, gibt es bereits eine lebhafte öffentliche Debatte über dieses Thema. Forscher, Politiker, Unternehmen sowie Interessenverbände von Konsumenten und Bürgerrechtlern fordern die zuständige Regulierungsbehörde Federal Communications Commission (FCC) auf, die schon verloren gegangene Neutralität des Netzwerkes durch Regulierung der Netzbetreiber wiederherzustellen.

Kontakt: Dr.-Ing. Barbara van Schewick, Fachgebiet Telekommunikationsnetze der TU Berlin, Fakultät IV, Sekr. FR 5-1, Franklinstr. 28/29, D-10587 Berlin, Tel. 030.314-23825, Fax: -314-23818, eMail: schewick@tkn.tu-berlin.de.

TU-Wissenschaftlerin Dr.-Ing. Barbara van Schewick warnt vor Eingriffen in den offenen Zugang des Internets.
Foto: TU Berlin